LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-AM-210388

Software Management
Quality Implementing Procedure ID: OSTI-LLNL-QIP-SI.0, Rev.0, Mod.0

V. J. Barish, L. A. Gouveia

March 10, 2005

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

SOFTWARE MANAGEMENT

Quality Implementing Procedure ID: OSTI-LLNL-QIP-SI.0, Rev. 0, Mod. 0 *Effective:* 2/25/05

1. PURPOSE

This Quality Implementing Procedure (QIP) establishes the responsibilities and processes for implementing and documenting activities that constitute software management in accordance with the OSTI-LLNL Quality Assurance Plan (QAP) which implements the U.S. Department of Energy (DOE) Office of Civilian Radioactive Waste Management (OCRWM) *Quality Assurance Requirements and Description* (QARD), DOE/RW-0333P, Supplement I. These activities include software planning, requirements, design, implementation, validation, operations and maintenance. This procedure provides for a centralized control of quality-affecting software and qualification documentation that is subject to the Office of Science & Technology and International (OSTI)-Lawrence Livermore National Laboratory (LLNL) Project.

Supplemental internal organizational controls (status accounting system, baseline software library, internal developmental guidelines, etc.) are allowed under this procedure specifying LLNL development controls that are necessary to ensure the life cycle phase activities delineated in this procedure are met.

2. SCOPE

This QIP applies to individuals within the OSTI-LLNL Project, and other participants who develop, use and administer software in support of OSTI-LLNL activities. This procedure applies to all software acquired or developed for the OSTI-LLNL Project. This QIP has been prepared in accordance with OSTI-LLNL-QIP-5.0, *Preparing the Quality Assurance Plan and Quality/Technical Implementing Procedures.*

Software items subject to the OSTI-LLNL QA Program requirements cannot be used in quality-affecting activities prior to the software being qualified and baselined. All source code developed or acquired for use solely with OSTI-LLNL resources must be submitted to OSTI-LLNL Software Configuration Management (SCM) in accordance with this procedure.

Qualified software accepted and obtained from the centralized Yucca Mountain Project (YMP) baseline for use by OSTI-LLNL has been determined to be acceptable and shall be noted in the OSTI-LLNL Status Accounting System as "YMP Software". Software Configuration Management for YMP Software is provided through procedures and with resources external to OSTI-LLNL. Any changes or revisions to an OSTI-LLNL baselined item shall be processed in accordance with Section 3.8 of this procedure.

2.1 Exemptions

The following types of software may be exempted from this procedure except as noted. The Project Manager (PM)/Deputy Project Manager (DPM) may elect to invoke the requirements of this procedure in lieu of exemption.

A. Software such as operating systems, system utilities, compilers and their associated libraries, office automation systems such as word processors, spreadsheets, e-mail, presentation software, desktop database managers. However, any applications developed using these types of software that are directly used for quality-affecting work must meet the exemption of 2.1.E below or be qualified according to this procedure.

B. Software used solely for visual display or graphical representation of data used in a product, or that controls data acquisition equipment but does not perform a calculation or obtain a solution, which is checked and approved in accordance with applicable procedures, and meets stated acceptance criteria. However, any applications developed using these types of software that are directly used for quality-affecting work must meet the exemption of 2.1.E below or be qualified according to this procedure.

C. Software used in scoping, feasibility studies, prototype development, or similar activities prior to actual use in quality-affecting activities. Such software shall be qualified in accordance with this procedure prior to use in quality-affecting activities.

D. Software that is embedded in the equipment or integral to the operations, maintenance, or calibration of measuring and test equipment and has not been developed or modified by the Software User. This type of software is tested and controlled in accordance with OSTI-LLNL-QIP-12.0, *Control of Measuring and Test Equipment and Calibration Standards*

E. Computations performed using the standard functions of a commercial off-the-shelf software program, whose results are not dependent on the software program itself and are documented sufficiently in a technically reviewed document such as a Technical Report, Model Report, or scientific notebook. "Documented sufficiently" means that the results can be independently reproduced or checked by hand calculations without recourse to the originator.

2.2. Software Subject to this Procedure

All software used for quality-affecting activities, other than software specified in Section 2.1, shall be qualified and documented in accordance with this procedure.

3. PROCEDURE

This section provides a systematic approach to software development and maintenance, allowing it to proceed in a traceable, planned, and orderly manner. All software shall be developed in a

manner that will provide for repeatable installation and operation on any computer that meets the requirements of the Target Platform(s) specified within the software documentation without recourse to the originator. Attachment 1, Software Management Flowchart, is a descriptive flow chart of this procedure and illustrates the process for obtaining and qualifying software. Section 6.0, Acronyms and Definitions, provides a list of acronyms and definitions used within the procedure.

3.1 Use of Qualified Software

A Software User shall refer to the Software Baseline Report available from OSTI-LLNL SCM to determine if qualified software is available to meet the identified need. If needed software is not available on the baseline, the PM/DPM shall designate a person to initiate a software plan.

If baseline software is available, the Software User initiates a Software User Request (SUR), Attachment 3 to request the software from SCM. The SCM Coordinator sends the requested installation package to the User. The PM/DPM designates an Installer, who may be the Software User, to install the software and perform the installation test(s) required to ensure that the software has been properly installed. The Software User shall then perform the activities needed to utilize the software for the intended purpose. The Software Problem Reporting and Resolution Process (Section 3.10) is followed if installation or usage errors are identified.

3.1.1 Requesting Qualified Software

A. **Software User:**

1. Locate desired software on the OSTI-LLNL Software Baseline Report and prepare an SUR in accordance with Attachment 3.

2. If software is not on the OSTI-LLNL Software Baseline Report, search the YMP Software Baseline Report for availability of a suitable code, request and install the software in accordance with the YMP procedures in effect.

3. Submit the SUR to OSTI-LLNL SCM to request distribution of or access to a baseline software item.

4. If qualified software is not available, proceed to Section 3.2.

B. **SCM Coordinator:**

1. Update the OSTI-LLNL SCM Status Accounting System.

2. Forward the requested software package (including the media and required user documentation) to the User or designated Installer.

C. **Installer:**

1. Install the software on the target platform in accordance with the installation instructions in the baselined documentation (e.g., Users Manual, LEVEL B Software Management Report [SMR]).

2. Confirm proper installation and software operation to SCM Coordinator by signing and dating the SUR. Provide a copy of the SUR to the software User and return the installation package and the updated SUR with supporting documentation to the Software Coordinator.

3. Report installation or operation failures using the Software Problem Reporting and Resolution process in Section 3.10 with notification to the PM/DPM and SCM Coordinator.

D. **SCM** Coordinator:

1. Update the OSTI-LLNL SCM Status Accounting System.

2. Submit the completed SUR and supporting documentation to the Records Coordinator maintenance in the Records Center (RC) per Section 4.0.

3.1.2 Use of Released Software

All qualified software being used must be obtained from OSTI-LLNL SCM, or through the appropriate YMP software procedure. This section addresses the controls that are required to be applied to qualified software being in use.

A. **Software User:**

1. Control and document the use of software so that comparable results can be obtained (with any differences explained) through independent replication of the activity or process within the range of validation to which the software was originally qualified. To accomplish this, the documentation must record, at a minimum, the use of the software within the Technical Report, Model Report, or Analysis Report in accordance with OSTI-LLNL-QIP-SIII.1, *Technical Reports,* and OSTI-LLNL-QIP-SIII.2, *Model Reports*, and OSTI-LLNL-QIP-SIII.5, *Scientific Analyses*, including the software name, version, operating environment, and range of use.

2. Qualify for use, in accordance with Section 3.8.3, any software used outside the documented range of validation. Software obtained from SCM used outside the range of validation shall be considered unqualified.

3. Refer to Section 3.8.4 of this procedure if the user's operating environment has been upgraded, or the computer replaced, after the software was obtained from SCM. Software used on a Target Platform different from that for which it was qualified is considered to be beyond the range of qualification.

3.2 Software Classification

Software developed by OSTI-LLNL Tasks will be classified as: (1) Level A software or (2) Level B software. Classification of software is determined based on how critical the nature of the software's functional and performance requirements are, and the complexity involved in completion of the software life cycle described in Section 3.3 of this procedure. Level A requires full software life cycle documentation, and Level B prescribes the minimum level of documentation to meet the requirements of this procedure. For Level B software, the requirements and design phase of the software life cycle are condensed into a single Software Management Report (SMR). However, as deemed necessary by the Deputy PM, a higher level of classification may be prescribed in the software plan.

Level A Software addresses one or more of the critical factors listed below. Level B Software is not required to meet any of these critical factors:

- The software is critical to the overall OSTI-LLNL mission, or involves a high level of risk.
- The software is highly complex, incorporates new technologies, or has stringent performance requirements.

Software developed for the purpose of pre- or post-processing of data that can be hand checked or visually inspected for accuracy may be considered to be Level B even though it falls in one or more of the above categories.

3.3 Software Life Cycle

The software life cycle consists of the following phases: requirements, design, implementation, validation testing, installation and checkout, operations and maintenance, and retirement. The number of phases and relative emphasis placed on each phase of software development will depend on the criticality of and level of effort involved in the software activity. Software life cycle activities may be performed in an iterative or sequential manner.

If the software is being developed to operate in more than one operating environment, (i.e., multiple platforms or using different operating systems), then this must be stated in the planning documentation and appropriate tests for each Target Platform must be included in the Installation Test Plan (ITP), the Validation Test Plan (VTP), and documented results summarized in the Validation Test Report (VTR).

Further guidance and information, (e.g., annotated outlines, review criteria, points of contact) is available from the SCM Coordinator.

3.3.1 Planning

The plan addressing software quality assurance for software development activities is prepared by completing a Software Configuration Control Request (SCCR – Attachment 2) and has the following elements:

A. A description of the overall nature and purpose of the software
B. The software products to which it applies
C. The organizations responsible for performing the software development work and achieving software quality, and their tasks and responsibilities
D. Required documentation
E. Standards, conventions, techniques, or methodologies that shall guide the software activity
F. Required software reviews
G. Methods for error reporting and corrective action.

The above software quality assurance planning elements are documented in the SCCR.

3.3.2 Requirements Phase

This phase specifies the development of requirements to ensure clear and consistent translation of user needs into software functional capabilities prior to its development, modification, or acquisition.

Requirements Document (RD)

The RD must address key elements of the software plan, such as the work scope, objectives, and the items listed in Section 3.3.1. Included in these requirements are specifications as to the sensitivity of the information to be processed or stored by the software so that appropriate security measures may be implemented. Software requirements shall provide enough detail to either design the software or make an acquisition decision. A software requirement shall only be specified if its achievement can be verified and validated.

The following are elements to be addressed and documented, as applicable, when developing the RD. Further guidance for preparation of the RD is available from the SCM Coordinator.

A. Functionality
B. Performance
C. Design Constraints
D. Attributes
E. External Interfaces
F. Communication Requirements
G. References

3.3.3 Design Phase

The decision to acquire or develop software is confirmed during the design phase. Software that meets the requirements of the RD may be acquired as Commercial Off-The-Shelf Software (COTS) or located in libraries of external public domain software. When software is acquired, it is non-Q and must be qualified for use per this procedure.

Design Document (DD)

If the decision is made to develop software or to modify acquired software, the requirements are translated into a set of technical computer system design specifications, which are described in the Design Document. To ensure traceability, the DD must address the RD requirements. The design specifies the data structures, processes, interfaces, and procedures to the level of detail necessary to plan and execute the implementation, validation, and installation of the software project. For acquired software that is not modified, a DD is not required. However, the Validation Test Plan (VTP) and Installation Test Plan (ITP) are required components of the Design Phase for all software developed or acquired.

The following are elements to be addressed and documented, as applicable, when developing the DD. Further guidance for preparation of the DD is available from SCM Coordinator. The design shall be described in a manner that can be translated into code.

A. A description of the major components of the software design as they relate to the software requirements.
B. A technical description of the software with respect to the theoretical basis, mathematical model, control flow, data flow, control logic, and data structure
C. A description of the allowable or defined ranges for inputs and outputs.
D. System Inputs and Outputs
E. Generation of Design-Based Test Cases
F. System and User Interfaces
G. References.

3.3.4 Validation Test Plan (VTP)

The VTP reflects the requirements of the RD, as well as the design elements of the DD for developed or modified software. The VTP documentation shall include:

A. A description of the design based test cases, including steps to be performed and unique identifier for each test case (e.g., 1, 2, 3, or Test 01Q, Test 02Q, Test 03Q, etc.)

B. The acceptance/rejection criteria (qualitative and/or quantitative)

C. Instructions for executing the VTP test cases

D. A place to indicate test status that includes pass/fail and initials of the tester and date. (e.g., test log).

E. Methods used to demonstrate technical adequacy (functional requirements or output), such as:

- Analysis without computer assistance (hand calculation).
- Other project-validated computer programs
- Experiments and tests
- Standard problems with known solutions
- Comparisons to confirmed published data correlations.

3.3.5 Installation Test Plan (ITP)

The ITP describes the installation requirements of the software product. If an acquired Users Manual (UM) or other vendor documentation contains these items, they may be referenced. The ITP documentation shall include, as appropriate:

A. Hardware and software Environment. Pre-installation tasks (e.g., de-fragmentation of the hard drive, determination of disk storage capacity) to be performed on the target platform and detailed description of the pre-installation test

B. Installation procedure

- Instructions that contain an introduction, description of the user's interaction with the software, and a description of any required training.
- Transfer of software and data elements from the distribution media to the target platform(s)
- The tasks to be performed after the software transfer has been completed
- Setting initial operating conditions automatically or by manual instructions in the procedure

C. Installation test case(s) – data files, input and output data, and defaults to accurately confirm a correct installation.

D. Installation acceptance criteria

E. Anticipated errors and how the user can respond.

3.3.6 Implementation Phase

Implementation is accomplished by developing new software or modifying existing software. During the implementation phase, the DD for developed or

modified software is translated into machine-readable code. The source code or executables for developed or modified software are generated and documented during this phase and verified against the DD and the RD. Additionally, developed or modified source code is unit tested, iteratively, as the individual software modules are coded and integrated. The UM for acquired, developed, or modified software shall be verified and approved for use during this phase.

Users Manual (UM)

The UM shall provide instructions for using the software and shall include, or in the case of acquired software, be supplemented as appropriate to include, the following:

A. Purpose and scope

B. Description of user interactions

C. Any constraints and/or special instructions to the users

D. Input/output options

E. Data files, input and output data, defaults, and file formats

F. Description of the allowable and tolerable ranges for inputs and outputs

G. Test methodology and test frequency for Continuous Use Software (see Section 3.11), if applicable.

H. Anticipated errors and how the user can respond

I. The hardware and software environment

J. Description of required training

K. Available sample problems

L. Installation procedures.

3.3.7 Validation Testing Phase

Software validation activities shall be performed, documented, and verified at the end of the implementation phase to ensure that the software installs correctly and satisfies the requirements for its intended use. During this phase, acquired, developed, or modified software is exercised by executing the test cases in the approved ITP and VTP. The validation process ensures that the software product adequately and correctly performs all intended functions and does not perform any unintended functions, either by itself or in combination with other functions, which can degrade the entire system. The validation process combines the results of the validation and installation testing activities into one VTR. Successful completion of this phase demonstrates that the software

product is ready for operational use. Software validation of modifications to released software shall be subjected to regression testing to detect errors introduced during the modification of the software, to verify that the modifications have not caused unintended adverse affects, or to verify that a modified software still meets specified requirements.

Validation Test Report (VTR)

The following elements are to be addressed and, if applicable, documented in the VTR. Further guidance on validation testing and preparation of the VTR is available from SCM.

A. Software Identification including version number

B. Target Platform(s)

C. Independent Validation Tester identification

D. Identification of special tools and equipment used (e.g., type, nomenclature, model numbers, serial numbers)

E. Test results linked to the unique test case identifier from the VTP

F. Documentation of results of execution of the individual test steps within the ITP and VTP, indicating that the test was executed and the correct results were obtained, with the applicable generated output, attached to the VTR

G. Indication of pass/fail, initial, and date for each VTP test case step.

H. Description of any failure conditions, how they occurred, and their resolution

I. Documentation and justification for any remaining test exceptions or failures

J. Summary of unit testing performed, if applicable

K. Overall conclusion and general remarks (optional).

3.3.8 Operations and Maintenance Phase

This phase includes those steps necessary to release the software to users and to control changes to the baseline whether from software enhancements, corrections of errors and software problems, or changes in the operating environment. These steps are described in Section 3.7, Software Operations and Maintenance.

3.3.9 Retirement Phase

Retirement of software shall be handled as a change to the baseline status of the software and in the status accounting system maintained within SCM. Software is retired from the baseline upon a determination that it is no longer to be used within LLNL-OSTI. Retirement of software shall be handled as a change to the baseline status of the software, which remains available for verification, and in the status accounting system maintained by SCM. These steps are described in Section 3.7.2, Canceling or Retiring Software.

3.4 Development and Control of Level B Software

This section applies to software classified as Level B. The documentation consists of two documents: the SCCR and the Software Management Report (SMR) for Level B Software.

3.4.1 Software Planning

Developer:

Initiate the software plan by preparing a Software Configuration Control Request (SCCR) form (Attachment 2) in accordance with the form instructions, including the software level classification information and submit the SCCR to the Deputy PM for approval. Obtain a Software Tracking Number, Software Media Number, and SMR Number from SCM in accordance with Section 3.5.1.

3.4.2 Preparation of Software Management Report (SMR)

Developer:

A. Prepare the SMR using the following elements. If any of these elements are not applicable, a brief justification of non-applicability is required for each heading that is not applicable. Further guidance for preparation of the SMR is available from the SCM Coordinator.

1. Software Identification

- Software name and version
- Software Tracking Number
- LEVEL B SMR Number
- LEVEL B Software Media Number.
- The name and version identification of the operating environment under which the software was developed to the level of detail normally associated with the operating system employed and the

compiler used (e.g., Excel 97, Microsoft Access 2.0, Sun OS 5.5.1, FORTRAN 90, etc.)

2. Description and Testing

   Provide documentation that the software provides correct results for a specified range of input parameters. Ensure the following information, at a minimum, is provided:

   - Description and equations of mathematical models, algorithms, and numerical solution techniques employed, as applicable
   - Description of software, including target platform(s)
   - Instructions adequate for installation and execution of the software without recourse to the originator or originating organization
   - Description of test cases, along with any necessary execution instructions
   - Description of test results and verification method(s)
   - Range of input parameter values for which results were verified
   - Description of required training
   - Identification of any limitations on application.

3. Supporting Information

   - Directory listing of executable and data files
   - Computer listing of source code, if available
   - Computer listing of test data input and output, identifying software name and version number (may be handwritten if not automatic for acquired software).

4. References

   Provide a reference list of documentation used in the development of the SMR.

B. Clearly identify the software documentation as the verification copy (e.g., "verification" or "verification copy") on the document cover sheet and assign an alphanumeric draft number, such as 00A.

C. Submit the SMR and the installation media to the independent Technical Reviewer and Software Coordinator to verify the completeness, accuracy,

and consistency with the above outline, in accordance with Section 3.6, and proceed to Step 3.4.2 D, below.

D. Produce the final SMR and change the alphanumeric designator to a numeric designator (e.g., If the draft document designator is "00B", then the subsequent approval copy is "00").

E. Sign and date the cover sheet of the SMR indicating the SMR meets the mandatory requirements, and that independent verifications have been performed and any comments resolved.

F. Obtain the reviewers' names and signatures, and date the cover sheet as "Verified By."

G. Label the software media in accordance with OSTI-LLNL-SV.0 "Control of Electronic Data."

H. Complete the SCCR (Attachment 2) to request that the submitted documents be controlled.

I. Submit the SCCR and SMR along with the software media to SCM for processing and control.

J. Submit an SUR (Attachment 3) in accordance with Section 3.1.1 to establish user information for Level B software.

3.5 Development and Control of Level A Software

This section applies to software classified as Level A. The documentation consists of: the SCCR, the Control Point 1 package and the Control Point 2 package. The Control Point 1 package includes the requirements, design, installation and validation test plans. The Control Point 2 package includes the users manual, validation test report, source code or executable code, for developed or acquired software, respectively, and/or any alternative approaches to testing, and the software media.

Upon a determination by a software user that needed software is not available on the OSTI-LLNL baseline of qualified software, and is also not available on the YMP baseline, the PM/DPM is notified. The PM/DPM assigns a Developer to initiate the software planning process.

3.5.1 Software Planning

A. The **Developer** initiates the software plan by preparing a Software Configuration Control Request (SCCR) form (Attachment 2) in accordance with the form instructions, including the software level classification information and submits the form to the PM/DPM for approval. If the SCCR is not approved, the PM/DPM notifies the Developer, and the process is discontinued.

B. The **PM/DPM** approves the SCCR by completing the PM/DPM actions in accordance with the form instructions, and returns the approved SCCR to the Developer who submits the SCCR to SCM Coordinator.

C. The **SCM Coordinator** assigns the Software Tracking Number (STN) and software media and document numbers on the SCCR, and updates the Status Accounting System with the SCCR information, and provides the Developer with the STN and document identifying numbers.

D. The **Developer** performs software life cycle activities in accordance with the software plan documented in the approved SCCR, and ensures that STN, software media number, and document identifying numbers are used on required documentation and media throughout the software development activity.

3.5.2 Control Point 1 Configuration Package

The Control Point 1 configuration package consists of five separate elements: SCCR, RD, DD, VTP, and ITP.

A. Requirements Analysis

Developer:

Perform a requirements analysis and prepare the RD in accordance with Section 3.3.2 of this procedure. If necessary, revise the descriptive sections in the SCCR to maintain traceability and consistency between the SCCR and the requirements.

B. Determination to Acquire, Develop, or Modify Software

During the design phase, the decision to acquire software is confirmed when software that meets the requirements of the RD may be acquired as Commercial Off-The-Shelf Software (COTS) or located in libraries of external public domain software. If the requirements of the RD cannot met, new software is developed or acquired software modified.

The design document is not required for acquired unmodified software.

C. Complete Preparation of the Control Point 1 Configuration Package

Developer:

1. Prepare the DD (if applicable) in accordance with Section 3.3.3, the VTP in accordance with Section 3.3.4, and the ITP in accordance with Section 3.3.5, of this procedure.

2. Identify the Control Point 1 documents as the verification copy (e.g., "verification" or "verification copy") on the document cover sheets and assign an alphanumeric draft number, such as 00A.

D. Control Point 1 Verification Review

Developer:

1. Submit the Control Point 1 configuration package to the Independent Technical Reviewer and Software Coordinator (or designee) for a verification review in accordance with Section 3.6. Once the review process is complete, proceed to Step 3.5.2.

2. Produce the final Control Point 1 documents by changing the alphanumeric designator to a numeric designator (e.g., if the draft document designator is "00B," the subsequent approval copy is "00," etc.).

3. Sign and date the Control Point 1 document cover sheet indicating that independent verifications have been performed and any comments resolved

4. Obtain the independent Technical Reviewer's name and dated signature on the cover sheet as "Verified By."

5. Obtain the Software Coordinator's (or designee) name and dated signature on the cover sheet as "Verified By."

6. Prepare the SCCR (Attachment 2) by completing the appropriate blocks on Page 2 for Control Point 1.

7. Submit the Control Point 1 documentation and the SCCR form to the SCM Coordinator for processing and subsequent record package compilation.

8. Retain a copy of the Control Point 1 documentation for use in Section 3.5.3, Control Point 2, Software Implementation.

E. SCM Coordinator:

1. Complete the SCCR form, Attachment 2, by completing the appropriate blocks.

2. Update the status accounting system.

3. Collect and submit the Control Point 1 documentation records and SCCR to the Records Coordinator per Section 4.0.

3.5.3 Control Point 2 Configuration Package

The Control Point 2 configuration package consists of the following, as applicable: SCCR, Users Manual (UM), source code, executables, installation, test and execution user instructions, read-me files describing media content, and the Validation Test Report (VTR).

Validation Testing

The installation and validation tests are performed as independently as possible from the development environment (i.e., use of a different machine). At a minimum, software must be uninstalled from the development machine and re-installed according to the ITP installation test planning documentation by an Independent Validation Tester. If the development machine is used for validation testing, appropriate safeguards (e.g., back-up copy of the code) shall be taken prior to uninstalling to ensure that the development environment can be restored to its original operational state

A. Developer:

1. Develop or modify source code in accordance with the Control Point 1, design, ensuring specifications have been met.

2. Perform unit testing on source code, as applicable, to ensure compliance with Control Point 1 and document the results for inclusion in the validation test results section of the (VTR).

3. Develop UM or supplement the vendor's UM by including special user instructions in accordance with Section 3.3.6 of this procedure.

4. Develop and prepare the software media in accordance with OSTI-LLNL-QIP-SV.0. At a minimum, the media must contain the read-me files that describe media content, executables, and all files necessary to successfully install and operate the software.

B. **Deputy Program Manager**:

Designate an independent Validation Tester(s) to conduct the installation and validation tests.

C. **Developer**:

Assign an alphanumeric draft number, such as 00A to the draft VTR. Provide copies of the ITP, VTP and in-process software media to the Validation Tester, together with a draft VTR.

D. **Validation Tester**:

1. Install the software in accordance with the ITP and run the tests described in the VTP and draft VTR.

2. Notify the Developer if any test cases cannot be executed, produce incorrect results, or that additional tests are needed to complete the VTP.

3. Complete the draft VTR in accordance with Section 3.3.7 of this procedure by initialing and dating the VTR to indicate that each specified test was conducted and produced acceptable results.

E. **Developer:**

1. If the installation or validation test fails based upon the Validation Tester's conclusions, identify and analyze the problems for correction in accordance with Section 3.8.2.

2. If the validation tests pass based upon the Validation Tester's conclusions, incorporate the test results into the validation test results section of the VTR, identify the VTR as the verification copy (e.g., "verification" or "verification copy") on the document cover sheet, and assign an alphanumeric draft number, such as 00A.

3.5.4 Control Point 2 Documentation Control

A. Developer:

1. Submit the Control Point 2 documents to the Independent Technical Reviewer and Software Coordinator for a verification review in accordance with Section 3.6. Once the review process is complete, proceed to Step 2.

2. Produce the final Control Point 2 package by changing the alphanumeric designator to a numeric designator (e.g., if the draft document designator is "00B," the subsequent approval copy is "00").

3. Sign and date the cover sheet of the Control Point 2 documentation indicating that independent verifications have been performed and any comments resolved.

4. Obtain the independent Technical Reviewer's name and dated signature on the cover sheet as "Verified By."

5. Obtain the Software Coordinator's (or designee) name and dated signature on the cover sheets.

6. Prepare the SCCR form, Attachment 2 by completing the appropriate blocks in Section II for Control Point 2.

7. Submit records of the Control Point 2 documentation, along with the SCCR and software media to the SCM Coordinator for processing and subsequent record package compilation in accordance with Section 4.0 of this procedure.

8. Submit an SUR (Attachment 3) to the SCM Coordinator in accordance with Section 3.1.1 to establish initial user information for this software item.

B. SCM Coordinator:

1. Complete the SCCR form (Attachment 2) by completing the appropriate blocks.

2. Update the Status Accounting System.

   Collect and process the record copies of the Control Point 2 documentation, Software User Request, and SCCR forms in accordance with Section 4.0 of this procedure.

3.6 Documentation Verification

3.6.1 PM/DPM:

A. Assign a Technical Reviewer to verify the software documentation. The individual selected to perform these reviews needs to be independent of all prior development and testing activities for the current software development activity.

   If, however, the Developer is the only individual that can perform the independent verification, document an explanation of the lack of independence in a memorandum and include the memorandum in the applicable review.

Developer:

A. Provide the verification copy of the document reviewed, Review Record and Comment Sheet per OSTI-LLNL-QIP-6.1, and/or media to the independent Technical Reviewer for verification, along with review criteria listed in 3.6.2 and any specific instructions.

B. Provide the verification copy of the document reviewed, Review Record and Comment Sheet per OSTI-LLNL-QIP-6.1, and/or media the Software Coordinator in accordance with 3.6.3.

3.6.2 Technical Review

Technical Reviewer:

A. Verify the software documentation, using applicable criteria below, based on the Control Point level and software type:

   1. The LEVEL B SMR is complete, accurate, consistent, and meets the requirements of Section 3.4.2 of this procedure.

2. The RD is complete, accurate, and consistent with the approved SP and meets the requirements of Section 3.3.2 of this procedure.

3. The DD, if applicable, the ITP, and VTP elements of the Control Point 1 are complete, accurate, and consistent with the RD and meet the requirements of Section 3.3.3 of this procedure.

4. The software media and UM (or supplemental UM) of the Control Point 2 are complete, accurate, and consistent with the approved DD (if applicable) and meet the requirements of Section 3.3.6 of this procedure.

5. The VTR element of the Control Point 2 is complete, accurate, and consistent with the approved ITP and approved VTP, and meets the requirements of Section 3.3.7 of this procedure.

6. The document was prepared in accordance with this procedure and the software plan.

7. The content and discussion of scientific approach and/or technical methods are technically adequate, correct, and comply with requirements.

8. The inputs are appropriate, current, correct, and useable in the software activity.

9. The assumptions, constraints, bounds, or limits are explicitly stated in the software documentation. They are considered reasonable, and their impact on the results is described and the quality is assessed in the results and conclusions.

10. The analytical and design approach and results are compatible with the software objectives.

11. Potential interfaces or interactions with other technical work or software activities are adequately identified and documented, such as work performed in sequence or a product received from another organization.

12. The referencing is thorough, accurate, and complete, including appropriate project tracking numbers (e.g., records accession numbers, Technical Information Center Catalog numbers, and data tracking numbers).

13. The conclusions and results are reasonable compared to the inputs and processes.

B. Clearly and legibly identify the independent Technical Reviewer's name on the cover sheet and write comments on the Comment Sheet or Verification Copy per OSTI-LLNL-QIP-6.1.

C. Return the review documentation, and software media (if applicable) to the Developer.

3.6.3 Software Quality Assurance Review

SCM Coordinator (or designee):

A. Perform a completeness review of the software documentation and media for procedural compliance.

B. Clearly and legibly identify the reviewer's name on the cover sheet and write comments on the Comment Sheet or Verification Copy.

C. Return the review documentation to the Developer.

3.6.4 Final Review

Developer:

A. Resolve all comments with the reviewers and ensure resolutions are documented. Documentation of review comments and their resolutions shall be retained as part of the records package.

B. Bring unresolved comments to the successively higher level of management for resolution. Resolutions are documented per OSTI-LLNL-QIP-6.1.

C. Modify or supplement software documentation, as required, to incorporate comment resolution.

D. Provide the following to the Technical Reviewer and Software Coordinator for final concurrence:

1. Initial "verification copy".
2. Final concurrence copy.

3.6.5 Technical Reviewer:

A. Sign the Review Record per OSTI-LLNL-QIP-6.1, confirming that comment resolutions have been incorporated as agreed.

B. Forward signed and dated final concurrence copy of the software verification documentation to the Developer.

3.6.6 SCM Coordinator:

A. Confirm that comment resolutions have been incorporated as agreed.

B. Sign and date the SCCR form (Attachment 2), where appropriate, for Level B SMR, Control Point 1 review or Control Point 2 review.

D. Return a copy of the SCCR form to the Developer.

3.6.7 Developer:

Forward the completed documentation to the SCM Coordinator for baseline software control.

3.7 Software Operations and Maintenance

This section includes those steps necessary to release the software for use and to control changes to the baseline whether from software enhancements, corrections of errors and defects, or changes in the operating environment. Retirement of software shall be handled as a change to the baseline status of the software and shall be reflected in the SCCR and in the status accounting system maintained within SCM.

3.7.1 Baselined Software Control and Release

SCM Coordinator:

A Coordinate any administrative issues regarding the baseline request package with the Developer.

B. Release Level B software upon receipt of a final SMR and a completed SCCR form.

C. Release Level software for use only if Control Point 1 documentation has been baselined, and if Control Point 2 documentation and media have been completed and reviewed by the Software Coordinator (or designee) and are ready for concurrent baselining at the time of the release. Upon incorporation of the software media and software documentation into the software baseline, the baseline copy becomes the master copy.

C. Update the Status Accounting System.

D. Prepare and distribute a notification of the baseline activity.

E. Submit the completed software records package and individual records to the Records Coordinator per Section 4.0.

3.7.2 Canceling or Retiring Software

This section is applicable to cancellations of software activities that were begun at the Developer's request and after the SCM assigned identification numbers, or upon the retirement of baselined software.

A. **Developer:**

1. Obtain approval of the Deputy PM for the proposed cancellation or retirement.

2. Submit a SCCR form to the SCM Coordinator providing the rationale for cancellation of the software activity prior to baselining or for retirement of the specific version of the software in question after baselining.

B. **SCM Coordinator:**

1. Notify authorized users of pending software retirement and obtain concurrence. E-mail may be used.

2. Update the Status Accounting System.

3. Notify documented users of the disposition of the software.

4. If applicable, and there are no non-concurring documented users, remove the software item from the baseline.

5. Complete the SCCR form and submit the SCCR to the Records Coordinator per Section 4.0.

3.8 Changes to Software, Documentation, and/or Operating Environment

3.8.1 In-Process Documentation Changes

This section describes how to handle changes that impact the Control Point 1 package after the technical review, but prior to being brought under the SCM Coordinator control at Control Point 1, or changes to the Control Point 2 package after the independent technical and the SCM Coordinator review, but prior to the documentation and software being baselined and released at Control Point 2.

Developer:

The change described below utilizes the same verification and approval methodology as invoked for the original document. Indicate changes to the documentation using A and B, or C, of the following:

A. A black vertical line in the right margin of the page.

B. An explanation in the Change History Page clearly indicating which individual sections were revised, as applicable.

C. An explanation in the Change History Page indicating the document was revised because the changes were too extensive to use A and B above.

However, editorial corrections may be made as necessary without invoking any of the above change processes as long as the Developer's initials and dates such corrections. Submissions of documentation at the Control Points are required to have any editorial changes clearly marked, initialed, and dated.

3.8.2 Software Development Changes

This section describes how to handle changes that impact the Control Point 1 documentation after being brought under the SCM Coordinator control at Control Point 1, but prior to the documentation and software being baselined and released at Control Point 2.

Developer:

A. Determine if changes are necessary for any Control Point 1 baseline documents (i.e., SCCR, RD, DD [if applicable], VTP, ITP) and make the changes accordingly.

B. Identify any changes in the remarks section on the SCCR form and submit the updated baseline documents at Control Point 2.

3.8.3 Baselined Software Changes

Developer:

A. Evaluate the changes for impact on life cycle phase products (i.e., SCCR, RD, DD [if applicable], VTP, ITP, software code, UM, or VTR) or on SMR.

B. Initiate a new SCCR that is tailored to define the degree of changes to the documentation and testing required for this change. The documentation review and testing can be limited to those portions of the documentation and software affected by the change, as identified in the SCCR.

C. Enter the life cycle process according to the determination made in the previous step, and process the software accordingly.

D. Re-execute the test cases in the ITP, VTP, and UM documents or the appropriate ITP, VTP, and UM sections of the Control Point 2, for changes associated with the modifications in order to revalidate that the software is operating as expected in the new environment.

E. Document the results of testing in a VTR in accordance with Section 3.3.7. of this procedure.

3.8.4 Change in Operating Environment

Changes in operating environment within the same platform and operating system family, e.g., Sun SunOS 5.6 to Sun SunOS 5.7) usually do not affect the execution of qualified software. Successful execution of installation tests is sufficient to demonstrate that the qualified software can operate on the changed environment. If the software cannot be installed by following the defined installation instructions, or fails an installation test, more extensive testing is required (Section 3.8.3). The steps to be followed when the user's operating environment changes are listed below.

A. **Software User:**

1. Prior to running qualified software, check the Software Baseline Report to verify the software is qualified in the new environment. If it is, submit an SUR to SCM for the software version and new environment in accordance with Section 3.1.1.

2. If the software is not qualified in the new environment, run the ITP cases in the new environment. If the ITP tests pass, submit an SUR for the software version and new environment, and an SCCR form documenting the ITP execution, to SCM. If the tests fail, notify SCM.

B. **SCM Coordinator:**

1. Issue a new Software Tracking Number in accordance with 3.5.1 when an SCCR form (Attachment 2) is received.

2. Process the SUR (Attachment 3).

3. If a user notifies SCM that qualified software could not be successfully installation-tested within a new environment, notify the developer that the software must be qualified in accordance with Section 3.8.3.

3.8.5 Changes Requiring Reinstallation

Examples of changes requiring reinstallation are when a computer malfunctions and is replaced with equivalent hardware or when software is moved for administrative purposes to an equivalent environment (same computer type and operating system) that has been previously qualified.

Developer:

Conduct reinstallation in accordance with Section 3.1.1 of this procedure.

3.9 Software Previously Developed Not Using This Procedure

This section shall apply only to unqualified software in which the history of the software is unknown, but the software is required to be used in quality-affecting work.

PM/DPM:

A. Determine the importance of the software for the continuing support of quality-affecting work.

B. Qualify software supporting quality-affecting work beginning with Section 3.2.

3.10 Software Problem Reporting and Resolution

This section describes the steps to be performed upon discovery of an error or defect in any software item controlled by this procedure or upon notification from an outside supplier that an error or defect exists in their product.

3.10.1 Software User:

A. Note the reproducibility of the problem (whether constant or intermittent). If the problem notification originated from outside of OSTI-LLNL, attempt to replicate the supplier-identified problem in the local environment.

B. Determine, in conjunction with the developing organization or supplier, if the problem is an actual software error or defect and not user induced. If the problem is determined to be user induced, stop the process.

C. Document the problem and the environment under which it occurred on a Software Problem Report (SPR), Attachment 4 in accordance with the instructions. If possible, print the screen with the error message and submit a copy with the SPR.

D. Attach supplier notification/documentation if the problem was discovered from a source outside of the program.

E. Submit the SPR to SCM.

3.10.2 SCM Coordinator:

A. Remove software from baseline until Step 3.10.3C is complete.

B. Identify the affected authorized Users.

C. Send copies of the SPR to the PM/DPM and to any identified users of the software. Inform them of the software suspension from the baseline.

D. Coordinate notification to the supplier of acquired software, if applicable.

E. Provide a copy of the SPR (Attachment 4) and any supporting documentation to the Developer (or designee).

3.10.3 Developer (or designee):

A. Conduct an evaluation of the impact of the problem, error or defect. The evaluation documentation must state one of the following conclusions:

- No impact exists (with detailed rationale for the determination of no impact)

- Insignificant impact exists (with detailed rationale for the determination of insignificance)

- Conditions exist that are adverse to quality and require initiation of Condition Report under OSTI-LLNL-QIP-16.0, *Condition Reporting and Resolution.*

B. If any of the output, generated, or developed data from the software have submitted to the Technical Data Management System (TDMS) and are adversely affected, notify the Technical Data Coordinator of the problem and its effect on ongoing and previous quality-affecting work.

C. Submit the evaluation documentation and the complete SPR form to SCM.

3.10.4 Developer (or designee):

A. Determine whether initiation of a modification to the software and/or accompanying baseline documentation is appropriate if the error or defect for the developed or acquired software is not determined to affect ongoing or previous quality-affecting work.

B. If modification is appropriate, begin in accordance with Section 3.8 of this procedure.

3.10.5 SCM Coordinator:

Submit the completed SPR form to the RC in accordance with Section 4.0 as part of the complete records package compilation.

3.11 Continuous Operation Software

Software that is acquired or developed to perform continuous data acquisition or process control functions shall have additional in-use tests in order to provide confirmation of correct results of the software.

3.11.1 Software User:

A. Periodic (as determined by the Developer) manual or automatic self-check, in-use tests shall be defined and performed for any software where computer errors, data errors, computer hardware failures, or instrument drift can affect the required performance. Describe the test methodology and test frequency in the software documentation and/or the UM.

B. Document and verify with reliable evidence that the tests have been conducted and the software is still operating as designed and implemented. Document test results in the Scientific Notebook or other QA record. Documentation shall consist of the following elements:

- Date the test was conducted
- Person conducting the test
- Test cases used
- Results of the test.

C. Forward the original or a copy of any in-use test documentation to SCM.

D. If objective evidence is found that the software is not operating as designed and implemented, follow the procedure in Section 3.10, Software Problem Reporting and Resolution.

3.11.2 SCM:

Submit in-use test documentation to the RC in accordance with Section 4.0 of this procedure as individual records.

3.12 Configuration Management

A software configuration management system will be established to include configuration identification, control, and status accounting. Software shall be placed under configuration management control as each baseline element is approved.

SCM Coordinator:

A. Establish configuration identification that includes:

1. The baseline elements for each software baseline and a unique identifier for each software item, including version or revision, to be placed under software configuration management.

2. Assignment of unique identifiers that relate baseline documents to their associated software items and maintain cross-references between baseline documents and associated software.

B. Maintain a configuration control system that includes:
(Note: Evaluation and approval of changes, software verification and validation is discussed in Section 3.8 of this procedure.)

1. A release and control process for baseline elements and changes thereto.

2. Formal control and documentation of changes to baseline elements. This documentation shall contain a description of the change, the rationale for the change, and the identification of affected baseline elements.

3. Ensure the changes are appropriately reflected in software documentation and document traceability is maintained.

4. Notify authorized users affected by the approved changes.

C. Establish and maintain a Status Accounting system that includes:

1. Unique software identifiers, the status of proposed, in-process, or approved changes to baselined elements, a brief chronology of the

software items, list of users, and a description of the changes made between versions of the software items.

2. Inform management and users upon request of the status through the various SCM reports that comprise the Software Configuration Status Accounting System.

4. RECORDS

Records listed in Section 4.1 shall be collected and submitted to the RC in accordance with OSTI-LLNL-QIP-17.0, *Records Management*, as individual records or included in a records package, as specified below. In the event the RC requires editorial corrections per OSTI-LLNL-QIP-17.0, the OSTI-LLNL Software Coordinator (or designee) is authorized to make required changes to the baseline documentation and shall initial and date each editorial correction.

4.1 QA Records

Records Package for Control Point 1:

- SCCR
- Requirements Document
- Design Document
- Installation Test Plan
- Validation Test Plan
- Review and Comment Sheets

Records Package for Control Point 2:

- Any modified Control Point 1 Documentation
- Validation Test Report
- Users Manual
- Review and Comment Sheets

Records Package for Level B Software:

- LEVEL B SMR, including SCCR
- Review and Comment Sheets

Individual Records:

- Completed Software Problem Report
- Completed SUR
- Documentation of error and defect evaluations
- Review and Comment Sheets
- In-use Test Documentation

Retired Software Records Package:

- SCCR

Operating Environment Change Records Package:

- SCCR
- SUR
- Supporting Documentation

4.2 Non-QA Long-Term Records

None.

4.3 Non-QA Short Term Records (three years or less retention)

None.

5. RESPONSIBILITIES

5.1 The **Project Manager (PM)/Deputy Project Manager (DPM)** is responsible for initiating the planning process for needed software that is not available on the software baseline, for assigning Developers, Technical Reviewers and Validation Testers for the software development life cycle, and for ensuring qualified software use. The DPM provides dated signature approval for all software control documentation, including cancellation and retirement from the baseline.

5.2 The **Developer** is responsible for software planning, design, development, software modifications, and test of operating system environment changes for qualified software and for documenting these activities in accordance with this procedure.

5.3 The **SCM Coordinator** is responsible for software quality management control and software configuration management in accordance with this procedure. The SCM Coordinator is responsible for reviewing the documentation and verification test results during the software life cycle for completeness, consistency, and conformity to this procedure, and to act as a point of contact for software questions, items, and issues.

5.4 The **Technical Reviewer** is responsible for review of software documentation and for determining whether or not the products(s) of a given phase of software development cycle fulfills the requirements imposed by the previous phases.

5.5 The **Validation Tester** is responsible for testing and evaluating developed software to ensure compliance with software requirements.

5.6 The **Installer** is responsible for installing qualified software on a computer in preparation for its use by a Software User.

5.7 The **Software User** is responsible for requesting needed software from the OSTI-LLNL Software Configuration Management baseline, maintaining a record of authorization for use, reporting software problems, and for controlling and documenting qualified software use within the range of validation and qualification.

6. ACRONYMS AND DEFINITIONS

6.1 Acronyms

CPU	Central Processing Unit
DD	Design Document
DOE	U.S. Department of Energy
IT	Information Technology
ITP	Installation Test Plan
ITR	Independent Technical Reviewer
LEVEL B	Software Management Report for Level B Software
LLNL	Lawrence Livermore National Laboratory
QA	Quality Assurance
QAP	Quality Assurance Plan
QARD	Quality Assurance Requirements Description
RD	Requirements Document
SCCR	Software Configuration Control Request
SCM	Software Configuration Management
SMR	Software Management Report
SPR	Software Problem Report
OSTI	Office of Science &Technology and International
UM	User Manual
VTP	Validation Test Plan
VTR	Validation Test Report
YMP	Yucca Mountain Project

6.2 Definitions

Acquired Software: Computer software obtained from an external source that was neither developed nor modified by the user.

Application Software: 1) A program or group of programs designed for end users. 2) Software that is written where the user prescribes one or more instructions to generate data, manipulates data, or performs calculations.

Baseline Element (Software): An individual software component (e.g., requirements document, design document, source code, etc.) that is under OSTI-LLNL configuration management control.

Baselined Software Changes: Alterations made to software that is under SCM control.

Baselined Software: Software that has been acquired, developed, or modified and that has been accepted by SCM for release and control, or has been deemed qualified pursuant to Section 2.0 hereof.

Commercial Off-The-Shelf Software (COTS): Software items that one can buy, ready-made, from some supplier's/retailer's store shelf or manufacturer's virtual store

shelf (e.g., through a catalogue or from a price list) on the basis of specifications set forth in the manufacturer's published product description (for example: a catalog or other published specification).

Control Point: A point at which the products of a series of life cycle phases are baselined and controlled.

Data Structures: A physical or logical relationship among data elements designed to support specific data manipulation functions.

Design Constraints: Limitations imposed on implementation activities. Any elements that will restrict design options.

Developed Software: Software that is developed or acquired software that has been modified.

Embedded Software: 1) The software associated with firmware that can provide function and control capabilities. 2) Software that is a fixed part of a larger system and performs some of the requirements of that system.

Installer: The individual installing the software on a User's computer in preparation for software use (normally as a result of a submitted Software User Request).

Modified Software: Baselined software that has been or is in the process of being changed (i.e., modifications to previously baselined developed software, previously baselined acquired software, or software that is acquired with the intent of modifying it prior to initial baselining.

Operating Environment: A collection of software, firmware, and hardware elements that provide for execution of computer programs. The operating environment includes the operating system, installed software applications, hardware, and configuration settings.

Operating System: A collection of software that controls the execution of computer programs and provides such services as computer resource allocation, job control, input/output, and file management in a complete system.

Qualified Software: Software that is acquired, developed, or modified that has successfully been included in the OSTI-LLNL Baseline of Qualified Software, or has been deemed qualified pursuant to Section 2.0 hereof.

Range of Validation: The documented type(s) of functionality and range of input for which the software is qualified.

Regression Testing: Selective retesting of a system or component to verify that modifications have not caused unintended effects, and that the system or component still complies with its specified requirements (QARD).

Software: Computer programs, procedures, rules, and associated documentation pertaining to the operation of a computer system (QARD).

Software Configuration Management: A discipline that applies technical and administrative controls for configuration identification, control, and status accounting.

Software Item: Source code, object code, job control code, control data, or a collection of these items that function as a single unit (QARD). For the purpose of this procedure software items also include the procedures, rules, associated documentation, and information pertaining to the operation of a computer system.

Software Life Cycle: A series of activities that begins when the software product is conceived and ends when the software product is no longer available for routine use (QARD). The software life cycle phases identified for this procedure are requirements, design, implementation (including coding, validation, installation and checkout, operations and maintenance, and retirement. The software development methodology enforced by this procedure is a combination of sequential and iterative methodology which allows iterative development to proceed, but requires that all applicable prior phases to be completed before any successor phase can be completed.

Software Medium: A device and/or entity (e.g., magnetic tape, diskette, CD-ROM, etc.) on which a software item is stored. Media shall be prepared using work activities/processes/process functions evaluated in accordance with OSTI-LLNL-QIP-SV.0

Software Problem: Any failure of a software document, code, data structure, or process to meet its requirements or standards.

Software Quality Assurance: A planned and systematic pattern of all actions required to provide necessary confidence that an item or product conforms to established requirements.

Software Validation: The test and evaluation of implemented software to ensure compliance with the requirements of this procedure.

Software Verification: The process of determining whether or not the product(s) of a given phase of the software development cycle fulfills the requirements imposed by the previous phase (QARD). An example of software verification is the comparison of the Design Document (DD) against the Requirements Document (RD) to ensure that all requirements identified in the RD are addressed in the DD.

Status Accounting: 1) Management of a centralized software baseline containing unique software identifiers, chronology of software, and a description of the changes made between versions of software. 2) Communication of software baseline status, including proposed, in-process, or approved software.

Target Platform: The Operating Environment in which the software is intended to be used.

Technical Report: As it pertains to scientific investigation, a document that presents scientific information such as data, analyses, interpretations, or conclusions (QARD).

Unit Testing: Testing of individual software units or groups of related units within a code that is performed during code implementation prior to final validation testing.

Validation Tester: The person technically qualified and independent of the software activity who is selected by the Deputy PM to perform Software Validation and Software Verification.

7. REFERENCES

DOE/RW-0333P, *Quality Assurance Requirements and Description*

OSTI-LLNL-QAP, *OSTI-LLNL Quality Assurance Plan*

OSTI-LLNL-QIP-5.0, *Preparing the Quality Assurance Plan and Quality/Technical Implementing Procedures*

OSTI-LLNL-QIP-12.0, *Control of Measuring and Test Equipment and Calibration Standards*

OSTI-LLNL-QIP-16.0, *Condition Reporting and Resolution*

OSTI-LLNL-QIP-17.0, *Records Management*

OSTI-LLNL-QIP-SV.0, *Management of OSTI-LLNL Electronic Data*

8. ATTACHMENTS

Forms attached to this procedure are controlled and are available from OSTI-LLNL SCM.

Attachment 1 - OSTI-LLNL Software Management Flowchart
Attachment 2 - Software Configuration Control Request
Attachment 3 - Software User Request
Attachment 4 - Software Problem Report

9. REVISION HISTORY

2/25/05 Revision 0, Modification 0

Initial issue.

10. APPROVALS

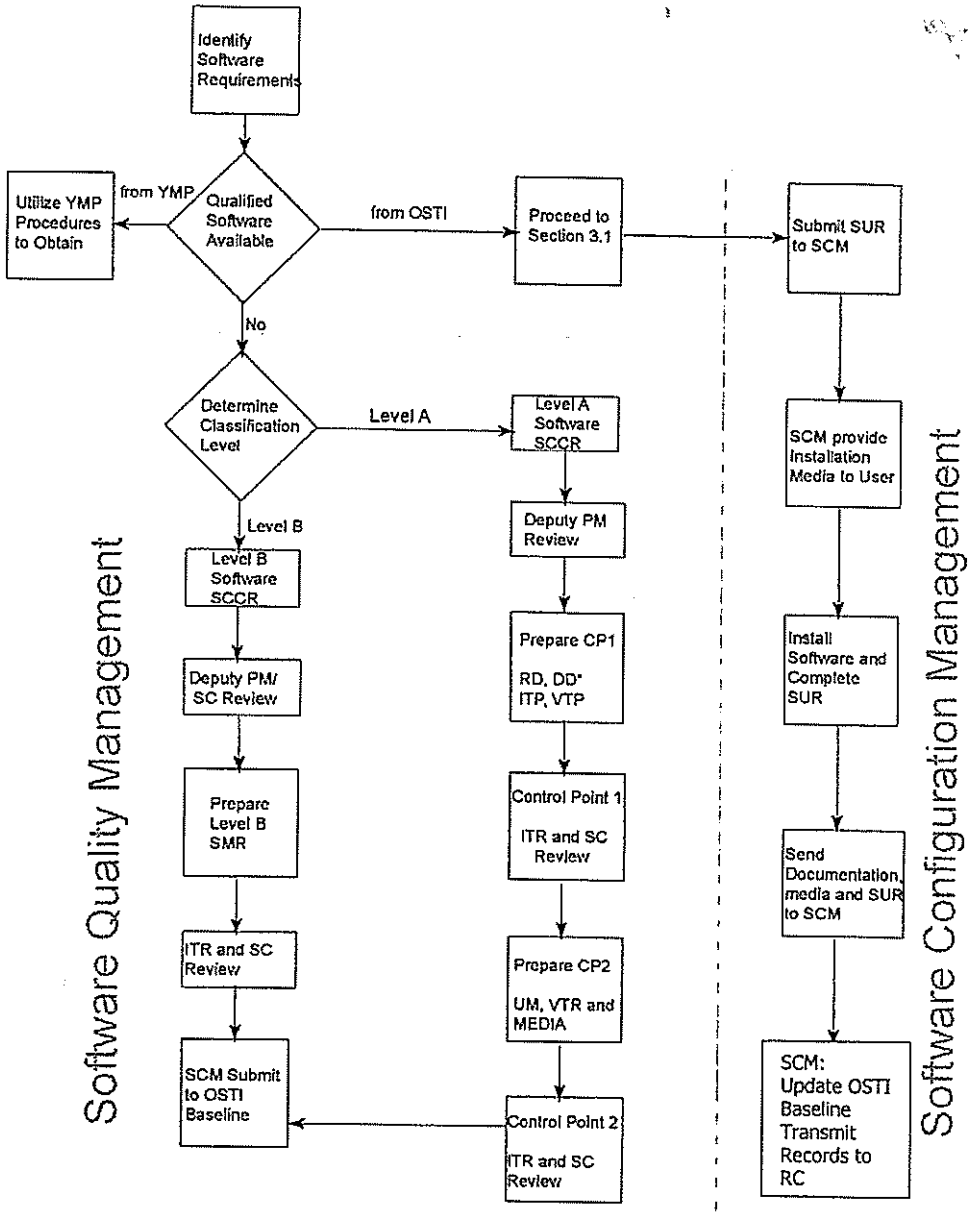
Preparer: Leigh Gouveia — Date: 2/25/05

Technical Reviewer: QINHONG HU — Date: 2/25/05

QA Reviewer: VICTOR J. BARISH JR — Date: 2/25/05

Project Manager: DAVID B. McCALLEN — Date: 2/25/05

OSTI LLNL Software Management Flowchart



Lawrence Livermore National Laboratory

OSTI-LLNL
Software Configuration Control Request

Complete only applicable item

QA:QA

Page 1 of :

1. Submission Activity

☐ New SCCR ☐ Modify in-process SCCR

2. Software Name	3. Software Version	4. Software Tracking Number (STN)

5. Software Activity

Initiate a new software activity
Change software on the baseline of qualified software
Retire qualified software
Cancel software activity
Change operating environment for qualified software

6. Overall Nature and Purpose of Software

EXAMPLE

Category

Level A ☐ Level B ☐

Continuous Use Software?

☐ Yes ☐ No

7. Software Life Cycle Methodology

☐ Iterative ☐ Sequential ☐ Hybrid of Iterative and Sequential

8. Standards, conventions, and techniques

☐ Generally accepted software engineering practices ☐ Other (describe below)

9. ☐ Required software reviews will be performed in accordance with OSTI-LLNL-QIP-SI.0.
Additional reviews planned:

10. ☐ Method for error reporting and corrective action will be performed in accordance with OSTI-LLNL-QIP-SI.0.
Additional error reporting and corrective actions planned:

11. Software Development Organization

12. Software Coordinator (Printed Name)	13. Software Coordinator's Organization	
14. Developer (Printed Name)	15. Developer Signature	16. Date
17. Project Manager/Deputy PM (Printed Name)	18. Project Manager/Deputy PM Signature	19. Date
20. SCM (Printed Name)	21. SCM Signature	22. Date Received 23. Date Entered

Lawrence Livermore
National Laboratory

OSTI-LLNL
Software Configuration Control Request

Complete only applicable items

QA: QA

Page 2 of :

2. Software Name	3. Software Version	4. Software Tracking Number (STN)

☐ Section I - Control Point 1, Requirements, Design, Installation and Validation Test Plan Baselining

☐ Section II- Control Point 2, Validation Test Report, Users Manual or Level B SMR Baselining

24. Select	25. Planned Documents	26. Document/Media ID	27. Comments
	Software QA Pan	OSTI-LLNL-QIP-SI.0, Rev.0, Mod. 0	
	Software Plan	This form.	
	Requirements Document (RD)		
	Design Document (DD)		
	Installation Test Plan (ITP)		
	Validation Test Plan (VTP)		
	Validation Test Report (VTR)		
	Users Manual (UM)		
	Software Media		
	Software Management Report (SMR)		
	Operating System Change Report		
	Evaluation for Acceptance Report		

28. Request that the following documents be controlled

29. Description and/or Rationale for the Submission Activity or Software Activity

30. Developer	Developer Signature	Date
31. SCM (Printed Name)	32. SCM Signature	33. Date Received
		34. Date Entered

Lawrence Livermore National Laboratory

OSTI-LLNL
Software Configuration Control Request

Complete only applicable items

QA: QA

Page 3 of :

2. Software Name	3. Software Version	4. Software Tracking Number (STN)

35. Description and/or Rationale for the Submission Activity or Software Activity

EXAMPLE

Instructions for Completing the OSTI-LLNL SCCR

Table of Contents

Page
Initiate a New Software Activity 5
Cancel Active SCCR 9
Retire Qualified Software 10
Change Qualified Software 11
Change in Operating Environment 15

Instructions for Completing the SCCR

Initiate a New Software Activity (Page 1 Instructions)

Developer

1. SUBMISSION ACTIVITY–Check "New SCCR" or "Modify in-process SCCR."

   "Modify in-process SCCR" should be used during the development activity if an additional document identifier is needed. For example:

   - The software is to be qualified on an additional environment, and another Validation Test Report document identifier for the added environment is needed.
   - The software has completed Control Point 1, and the Requirements Document needs to be revised. A new document identifier for ht revision to the Requirements Documented is needed.

2. SOFTWARE NAME–Enter the name of the software.

3. SOFTWARE VERSION–Enter version number of the software.

5. SOFTWARE ACTIVITY–Check "Initiate a new software activity." Check if "Verification copy" or "Final." Enter revision designation.

6. OVERALL NATURE AND PURPOSE OF SOFTWARE–Provide a high-level description of the software in terms of its functionality (i.e., what does it do?) and its use (i.e., what is it used for?).

   CONTINUOUS USE?–Check "Yes" if software will be continuous use software or "No" if software will not be continuous use software.

7. SOFTWARE LIFE CYCLE METHOLODOGY–Check the software life cycle methodology to be used in the software development process.

   - Iterative–Methodologies allow parallel execution of life cycle phases (e.g., requirements, design, and implementation).
   - Sequential–Methodologies require that serial execution of life cycle phase such that the commencement of one life cycle phase must follow the completion of the previous phase.
   - Hybrid of iterative and sequential treatment of life cycle phases–Note: this allows, for example, requirements, design, and implementation phases to be executed iteratively, but the testing phase to be executed only after the conclusion of the earlier phases.

8. STANDARDS, CONVENTIONS, AND TECHNIQUES–Check the appropriate box that describes the standards, conventions, and techniques that will guide the development process. If checking "Other," describe the standards, conventions, and/or techniques that are to guide the software development process.

   For many software development projects, the standards, conventions, and techniques that guide the software development process are generally accepted software engineering practices. However, other standards, conventions, and/or techniques may be specified.

   - Standard–An approved, documented, and available set of criteria.
   - Convention–A basic principle or procedure accepted as true or correct by general agreement.
   - Technique–A method of accomplishing a desired aim.

9. "Required software reviews will be prepared in accordance with OSTI-LLNL-QIP-SI.0." Indicate concurrence by checking box.

   ADDITIONAL REVIEWS PLANNED–Specify any additional reviews planned in the space provided. Use Page 3 if additional space is needed.

10. "Method for error reporting and corrective action will be performed in accordance with OSTI-LLNL-QIP-SI.0" – Indicate concurrence by checking box.

    ADDITIONAL ERROR REPORTING AND CORRECTION ACTION ACTIVITIES PLANNED–Specify any additional error reporting and corrective action activities planned in the space

provided. Use Page 3 if additional space is needed.

Instructions for Completing the SCCR (Continued)

Initiate a New Software Activity (Page 1 Instructions) (Continued)

11. SOFTWARE DEVELOPMENT ORGANIZATION–Enter the organizational code of the organization responsible for the development of the software (e.g.; Los Alamos National Laboratory [LANL]; Lawrence Berkeley National Laboratory [LBNL]; Lawrence Livermore National Laboratory [LLNL]; Sandia National Laboratories [SNL]; U.S. Geological Survey [USGS]).

12. SOFTWARE COORDINATOR'S NAME–Enter the name of the Software Coordinator.

13. SOFTWARE COORDINATOR'S ORGANIZATION–Enter the organization to which the Software Coordinator belongs. Use the organization code in Block 11 instructions.

14. DEVELOPER NAME–Type or print the name of the Developer.

15. SIGNATURE–Sign when the form is completed.

16. DATE–Enter the date of the signature.

PM/DPM

17. PM/DPM NAME–Type or print the name of the PM/DPM.

18. SIGNATURE–Sign when the software plan is approved.

19. DATE–Enter the date of the signature.

SCM

22. DATE RECEIVED–Enter the date received by SCM to assign STN and/or document identifier.

4. SOFTWARE TRACKING NUMBER (STN)–Enter the STN assigned to this software activity.

20. SCM NAME–Type or print the name of the Software Configuration Management individual completing the form.

21. SIGNATURE–Sign when the STN and/or document identifier are assigned.

23. DATE ENTERED–Enter the date of the signature.

Instructions for Completing the SCCR (Continued)

Initiate a New Software Activity (Page 2 Instructions)

Developer

2. SOFTWARE NAME–Enter the name of the software.

3. SOFTWARE VERSION–Enter version number of the software.

4. SOFTWARE TRACKING NUMBER (STN)–If a new SCCR, leave blank; this is entered by SCM. If a modification to an in-process SCCR, enter the STN assigned to this activity.

24. SELECT–Place a check by each product that will be produced during this activity, e.g.:

- Requirements Document
- Design Document
- Implementation Document
- Validation Test Report (a separate one for each operating environment in which the software is to be qualified.
- Distribution Media
- Operating System Change Report
- Evaluation for Acceptance Report.

25. PLANNED DOCUMENTS–In the blank row at the bottom of the list of planned documents, provide the title of any planned products in addition to those pre-identified by the form and select the corresponding box in Block 24. Limit one configuration item per row. If more than one operating environment is being qualified, list the operating environment in the Comment block for the associated Validation Test Report.

27. COMMENTS–For all products, provide any comments that will be helpful in processing the SCCR.

For each hardware/software environment on which the software will be qualified, provide the requisite components of that hardware/software environment on a separate Validation Test Report. If additional Validation Test Reports are required (beyond those pre-printed on the form), create additional Validation Test Reports using the instruction for Block 24. Typically, requisite components of hardware/software environments include the following:

- Central Processing Unit architecture (e.g., Intel x86, Sun SparC).
- Operating system (e.g., Microsoft Windows 2000, Sun SunOS 5.5.1)
- Operating system patches (e.g., Service Pack 4)
- Other hardware or software components not covered by the above

29. DESCRIPTION AND/OR RATIONALE FOR THE SUBMISSION ACTIVITY OR SOFTWARE ACTIVITY–Provide the rationale for adding the specified software to the baseline of qualified software.

SCM

4. SOFTWARE TRACKING NUMBER (STN)–If a new SCCR, enter the STN assigned to this software activity.

26. DOCUMENT/MEDIA ID–For each document checked other than "Software Quality Assurance Plan" or "Software Plan," enter the appropriate identifier.

33. DATE RECEIVED–Enter the date Control Point 2 package received from Developer.

31. SCM NAME–Type or print the name of the SCM individual completing the form.

32. SIGNATURE–Sign when the software is qualified.

34. DATE ENTERED–Enter the date of the signature.

Instructions for Completing the SCCR (Continued)

Initiate a New Software Activity (Page 3 Instructions)

Note:	Page 3 is optional.

Developer

2. SOFTWARE NAME–Enter the name of the software.

3. SOFTWARE VERSION–Enter version number of the software.

4. SOFTWARE TRACKING NUMBER (STN)–If a new SCCR, leave blank; this is entered by SCM. If a modification to an in-process SCCR, enter the STN assigned to this activity.

35. CONTINUATION PAGE–Identify the item being continued with the item number.

Instructions for Completing the SCCR (Continued)

Cancel Active SCCR

Developer

1. SUBMISSION ACTIIVTY–Check "New SCCR."
2. SOFTWARE NAME–Enter the name of the software for which the software activity is being cancelled.
3. SOFTWARE VERSION–Enter version number of the software for which the software activity is being cancelled.
4. SOFTWARE TRACKING NUMBER (STN)–Enter the STN of the software to be cancelled.
5. SOFTWARE ACTIVITY–Check "Cancel software activity."
29. DESCRIPTION AND/OR RATIONALE FOR THE SUBMISSION ACTIVITY OR SOFTWARE ACTIVITY–Provide rationale for canceling the software activity.
14. DEVELOPER NAME–Type or print the name of the Developer.
15. SIGNATURE–Sign.
16. DATE–Enter the date of the signature.

Deputy PM

17. DEPUTY PM NAME–Type or print the name of the Deputy PM.
18. SIGNATURE–Sign.
19. DATE–Enter the date of the signature.

SCM

22. DATE RECEIVED–Enter date received by SCM.
20. SCM NAME–Type or print the name of the Software Configuration Management individual completing the form.
21. SIGNATURE–Sign when Status Accounting is updated and the requested activity is complete.
23. DATE ENTERED–Enter the date of the signature.

Instructions for Completing the SCCR (Continued)

Retire Qualified Software

Developer

1. SUBMISSION ACTIVITY–Check "New SCCR."
2. SOFTWARE NAME–Enter the name of the software to retire.
3. SOFTWARE VERSION–Enter version number of the software to retire.
4. SOFTWARE TRACKING NUMBER (STN)–Enter the STN assigned to the software to be retired.
5. SOFTWARE ACTIVITY–Check "Retire qualified software."
29. DESCRIPTION AND/OR RATIONAL FOR THE SUBMISSION ACTIVITY OR SOFTWARE ACTIVITY–Provide a rationale for retiring the software.
17. DEVELOPER NAME–Enter the name of the Developer.
18. SIGNATURE–Sign.
19. DATE–Enter the date of the signature.

SCM

22. DATE RECEIVED–Enter date received by SCM.
20. Enter the name of the SCM individual completing the form.
21. SIGNATURE–Sign when Status Accounting is updated and the requested activity is complete.
23. DATE ENTERED–Enter the date of the signature.

Instructions for Completing the SCCR (Continued)

Change Qualified Software (Page 1 Instructions)

Developer

1. SUBMISSION ACTIVITY–Check "New SCCR" or "Modify active SCCR."

   "Modify in-process SCCR" is used during the development activity if an additional document identifier is needed. For example:

   - The software is to be qualified on an additional environment, and another Validation Test Report document identifier for the added environment is needed.
   - The software has completed Control Point 1, and the Requirements Document needs to be revised. A new document identifier for ht revision to the Requirements Documented is needed.

2. SOFTWARE NAME–Enter the name of the software.

3. SOFTWARE VERSION–Enter version number for the resulting software.

   It is the responsibility of the developer organization to establish the appropriate version number. The version on the baseline may be left unchanged for minor changes such as an editorial modification of a document or an addition of an operating environment that does not require modification to the source code or rebuilding/recompilation of the executable. However, the version number may be incremented for substantive changes such as modification of the software's functionality.

5. SOFTWARE ACTIVITY–Check "Change software on the baseline of qualified software."

11. SOFTWARE DEVELOPMENT ORGANIZATION–Enter the organizational code of the organization responsible for the development of the software (e.g.; Los Alamos National Laboratory [LANL]; Lawrence Berkeley National Laboratory [LBNL]; Lawrence Livermore National Laboratory [LLNL]; Sandia National Laboratories [SNL]; U.S. Geological Survey [USGS]).

12. SOFTWARE COORDINATOR NAME–Enter the name of the Software Coordinator.

13. SOFTWARE COORDINATOR'S ORGANIZATION–Enter the organization to which the Software Coordinator belongs. Use the organization codes in Block 11 instructions.

14. DEVELOPER–Type or print the name of the Developer.

15. SIGNATURE–Sign when the form is completed.

16. DATE–Enter the date of the signature.

Deputy PM

17. DEPUTY PM NAME–Type or print the name of the Deputy PM.

18. SIGNATURE–Sign when the software activity is approved.

19. DATE–Enter the date of the signature.

SCM

22. DATE RECEIVED–Enter date received by SCM to assign STN and/or document identifier.

4. SOFTWARE TRACKING NUMBER (STN)–Enter the STN assigned to this software activity.

20. SCM NAME–Type or print the name of the SCM individual completing the form.

21. SIGNATURE–Sign when the STN and/or document identifier are assigned.

23. DATE ENTERED–Enter the date of the signature.

Instructions for Completing the SCCR (Continued)

Change Qualified Software (Page 2 Instructions)

Developer

2. SOFTWARE NAME–Enter the name of the software.

3. SOFTWARE VERSION–Enter version number for the resulting software.

4. SOFTWARE TRACKING NUMBER (STN)–If a new SCCR, leave blank; this is entered by SCM. If a modification to an in-process SCCR, enter the STN assigned to this activity.

24 -27 Specify the documents that are planned to be changed. If a document is unchanged, annotate that in Column 26; use the instructions for "Initiate a New Software Activity."

30. DESCRIPTION AND/OR RATIONALE FOR THE SUBMISSION ACTIVITY OR SOFTWARE ACTIVITY–Provide a description and rationale for changing the software.

SCM

4. SOFTWARE TRACKING NUMBER (STN)–If a new SCCR, enter the STN assigned to this software activity.

26. DOCUMENT/MEDIA ID–For each document checked other than "Software Quality Assurance Plan" or "Software Plan," enter the appropriate identifier.

33. DATE RECEIVED–Enter the date the package was received from Developer.

31. SCM NAME–Type or print the name of the SCM individual completing the form.

32. SIGNATURE–Sign when the software is qualified.

34. DATE ENTERED–Enter the date of the signature.

Instructions for Completing the SCCR (Continued)

Change Qualified Software (Page 3 Instructions)

Note:	Page 3 is optional.

Developer

2. SOFTWARE NAME–Enter the name of the software.

3. SOFTWARE VERSION–Enter version number of the software.

4. SOFTWARE TRACKING NUMBER (STN)–If a new SCCR, leave blank; this is entered by SCM. If a modification to an in-process SCCR, enter the STN assigned to this activity.

35. CONTINUATION PAGE–Identify the item being continued with the item number.

Instructions for Completing the SCCR (Continued)

Change in Operating Environment (Page 1 Instructions)

Developer

1. SUBMISSION ACTIVITY–Check "New SCCR."
2. SOFTWARE NAME–Enter the name of the software.
3. SOFTWARE VERSION–Enter version number for the resulting software.
5. SOFTWARE ACTIVITY–Check "Change operating environment for qualified software."
29. DESCRIPTION AND/OR RATIONALE FOR THE SUBMISSION ACTIVITY OR SOFTWARE ACTIVITY–Provide a description of the new operating system environment version.
11. SOFTWARE DEVELOPMENT ORGANIZATION–Enter the organizational code of the organization responsible for the development of the software (e.g.; Los Alamos National Laboratory [LANL]; Lawrence Berkeley National Laboratory [LBNL]; Lawrence Livermore National Laboratory [LLNL]; Sandia National Laboratories [SNL]; U.S. Geological Survey [USGS]).
12. SOFTWARE COORDINATOR NAME–Enter the name of the Software Coordinator.
13. SOFTWARE COORDINATOR'S ORGANIZATION–Enter the organization to which the Software Coordinator belongs. Use the organization codes in Block 11 instructions.
14. DEVELOPER–Type or print the name of the Developer.
15. SIGNATURE–Sign when the form is completed.
16. DATE–Enter the date of the signature.

SCM

22. DATE RECEIVED–Enter date received by SCM to assign STN.
4. SOFTWARE TRACKING NUMBER (STN)–Enter the STN assigned to this software activity.
20. SCM NAME–Type or print the name of the SCM individual completing the form.
21. SIGNATURE–Sign when the STN and/or document identifier are assigned.
23. DATE ENTERED–Enter the date of the signature.

Instructions for Completing the SCCR (Continued)

Change in Operating Environment (Page 2 Instructions)

Developer

2. SOFTWARE NAME–Enter the name of the software.

3. SOFTWARE VERSION–Enter version number for the resulting software.

4. SOFTWARE TRACKING NUMBER (STN)–If a new SCCR, leave blank; this is entered by SCM. If a modification to an in-process SCCR, enter the STN assigned to this activity.

24.-27. Specify the Validation Test Report for the new operating environment that is planned to be qualified.

29. DESCRIPTION AND/OR RATIONALE FOR THE SUBMISSION ACTIVITY OR SOFTWARE ACTIVITY–Provide a description and rationale for changing the operating environment.

SCM

4. SOFTWARE TRACKING NUMBER (STN)–If a new SCCR, enter the STN assigned to this software activity.

26. DOCUMENT/MEDIA ID–For each document checked other than "Software Quality Assurance Plan" or "Software Plan," enter the appropriate identifier.

31. SCM NAME–Type or print the name of the SCM individual completing the form.

32. SIGNATURE–Sign when the software is qualified.

34. DATE ENTERED–Enter the date of the signature.

Lawrence Livermore
National Laboratory

OSTI-LLNL
Software User Request

QA:QA

Page 1 of :

Complete only applicable items

1 Software Name	2. Version	3. Software Tracking Number

4. Organizational and geographical location where software will be used:

5. Central Processing Unit (CPU) Number(s) (property tag number(s) or CPU serial number(s)/ Operating System:

6 State the intended use of the software:

EXAMPLE

7.

☐ Distribute Software to a Single Computer

8.

x Server Access

☐ Single User Access

☐ Read

☐ Write

☐ Multiple (Global) User Access

☐ Read

☐ Write

9. List of users who are qualified to use the software and will have access to the CPU(s):

10. Deputy Program Manager:	Organization:	Date:
Telephone Number:	Fax Number:	
11. SCM:		Date:

12. ☐ Installation/Reinstallation Successful. Software operating as expected.

Installer Name: Initials: Date:

INSTRUCTIONS FOR COMPLETING THE OSTI-LLNL SOFTWARE USER REQUEST

User:

1. Enter the software name.

2. Enter software version number.

3. Enter the previously assigned Software Tracking Number.

4. Enter the organizational and geographical location where the software will be used (e.g., LBNL, LLNL, etc).

5. Provide an appropriate identifying number such as an LLNL property tag number or computer serial number.

6. State the intended use of the software and describe the environment (platform, operating systems, etc.) in which the software will be used. Also indicate type of media needed for operation of this program in case multiple platform versions of this software exist.

7. Mark if the software is to be distributed and installed on a specific computer or set of computers.

8. Mark if the software is to be mounted on a server that can be accessed by multiple persons. Further indicate if this server access is for a single user (read or read/write) or if the software has been developed for global access (read or read/write).

9. List known users who will have access to this software.

10. Enter Deputy PM Name (signature is not required, Organization, enter the date, Telephone and Facsimile Number.

SCM:

11. Enter the page count on all pages of the Software User Request (including attachments, if applicable). Enter SCM name (signature is not required) and enter the date.

Installer:

12. After successful installation, check the box, enter Installer Name, initial, and date.

Lawrence Livermore
National Laboratory

OSTI-LLNL
Software Problem Report

Complete only applicable items

QA:QA

Page 1 of :

1. Software Tracking Number:	2. Software Name and Version:	3. SPR Number:

4. Source of Error or Defect:

☐ Internal to OSTI-LLNL

☐ External to OSTI-LLNL (Outside User)

Provide Contact Information (if available):

5. Date Problem Encountered:	6. Has Problem Been Verified as an Error or Defect: ☐ Yes ☐ No

7.:Description of the problem, the environment in which it occurred, and proposed resolution, if any.

EXAMPLE

8. User or Developer Name:	User or Developer Org.:	Date
9. Software Verifier Name and Signature:		Date:
10. SCM Name and Signature:		Date:

INSTRUCTIONS FOR COMPLETING THE OSTI-LLNL SOFTWARE PROBLEM REPORT

User:

1. Enter the previously assigned Software Tracking Number.

2. Enter the software name and version number.

3. Obtain SPR number from SCM.

4. Indicate whether the problem was internally discovered or came from an external source by checking the appropriate box. Provide supplier or outside user name, address, and phone (if known otherwise N/A).

5. Enter the date the problem was first discovered or made known to you.

6. Indicate if the problem has been verified (reproduced).

7. Provide a description of the problem. Give as much detail as possible about the problem and the operating environment in use (hardware, OS, commands issued, inputs, error messages, error logs, etc.). Include where in the software code (what module, if any) the problem occurred, and under what conditions, if known. Attach a copy of the notification from the supplier, if any, and reference the attachment in this block. Give your comments on any suggested resolution to this problem.

8. Enter User or Developer Name (signature is not required), Organization and enter the date.

SCM:

9. Software Verifier will review the SPR and verify the documentation. Fill out SCCR Form, page 3, requesting that software be removed from use until Developer has conducted an evaluation to include a statement of the impact of the error or defect. Attach the SCCR Form, page 3. Fill in the Software Verifier Name, sign and enter the date.

10. SCM Name and Signature. Ensure that the software has been removed from the Software Baseline Report for use for the project and enter the date.